Compression image sharing using DCT- Wavelet transform and coding by Blackely method

[1]Ali H. Ahmed [1], Loay E. George [2]

[1] College of Science / University of Baghdad
ali_ha_2012@yahoo.com

[2] College of Science / University of Technology
loayedwar57@yahoo.com

Abstract: ***The increased use of computer and internet had been related to the wide use of multimedia information. The requirement for protecting this information has risen dramatically. To prevent the confidential information from being tampered with, one needs to apply some cryptographic techniques. Most of cryptographic strategies have one similar weak point that is the information is centralized. To overcome this drawback the secret sharing was introduced. It's a technique to distribute a secret among a group of members, such that every member owns a share of the secret; but only a particular combination of shares could reveal the secret. Individual shares reveal nothing about the secret. The major challenge faces image secret sharing is the shadow size; that's the complete size of the lowest needed of shares for revealing is greater than the original secret file. So the core of this work is to use different transform coding strategies in order to get as much as possible the smallest share size. In this paper Compressive Sharing System for Images Using Transform Coding and Blackely Method based on transform coding illustration are introduced. The introduced compressive secret sharing scheme using an appropriate transform (Discrete cosine transform and Wavelet) are applied to de-correlate the image samples, then feeding the output (i.e., compressed image data) to the diffusion scheme which is applied to remove any statistical redundancy or bits of important attribute that will exist within the compressed stream and in the last the (k, n) threshold secret sharing scheme, where n is the number of generated shares and k is the minimum needed shares for revealing. For making a certain high security level, each produced share is passed through stream ciphering depends on an individual encryption key belongs to the shareholder.***

1- INTRODUCTION

Today, a secret image is transferred over the internet for numerous commercial purposes; therefore it is necessary to make sure information isn't being tampered [1]. Several cryptographic protocols and schemes were designed to solve the issues of this kind. Secret sharing schemes are important tools in cryptography and they are used as building boxes in several secure protocols [2], [3].

Secret sharing scheme starts with a secret and then derives from it certain shares (or shadows) which are distributed to users. So, those only qualified subsets are able to recover the information. The individual shares should reveal nothing about the secret [4], [5].

The security of any secret sharing scheme is measured by how much data about the secret is given by each one of shares and how much is given to a group of shares. The efficiency and security of a system decrease as the quantity of the data that has to be kept secret increases [4]. A secret sharing scheme is considered ideal approach once the threshold shares give absolutely no data regarding the hidden secret [6].

Blakley secret sharing scheme has an approach based on hyperplane geometry. To implement a (t, n) threshold scheme, each of the ***n*** users is given a hyperplane equation in a ***t*** dimensional space over a finite field such that every hyperplane passes through a particular point. The intersection point of the hyperplanes is the secret. When n users come together, they will solve the system equations to find the secret [7].

The ***(k, n)*** threshold algorithm relies on Blakley's secret sharing methodology. It uses the principle of hyperplane geometry to resolve the secret sharing problem. According to this scheme, the secret is a point in ***k*** dimensional space which is the point of intersection of all the hyperplanes. The affine of a hyperplane in a ***k*** dimensional space will be represented in ***n*** shares. Blakley's SSS will be represented using matrix notation [18]:

$$\begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & \dots & a_{2k} \\ a_{31} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & \dots & a_{Nk} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_k \end{pmatrix} mod\ p \qquad (1)$$

Or equivalently,

$$s_i = \left(\sum_{j=1}^{k} a_{ij} V_j \right) = mod\ p\ \ (for\ 1 \le i \le k) \qquad (2)$$

Where, $\{aij \mid j=1, 2\dots k\}$ is the set of sharing coefficients for generating the i^{th} share; $\{Vj \mid j=1, 2... k\}$ is the set of secret, p is the prime number (e.g., 257, 131, 67, 37 or 17). The set of a's components values should be chosen such that all components satisfy the following condition:

$$(a_{i1}\, a_{2k} - a_{i2}\, a_{1k})\, mod\ p \neq 0\ for\ \forall i \neq k \qquad (3)$$

The revealing part of (k, n) threshold scheme is the inverse of the sharing part. The revealing step simply finds the solution for a set of linear equations. The secret bytes $\{V_i \mid i=1, 2, \dots, k\}$ will be obtained by finding the following equation that represents the general solution equation for the share equation (3.15):

$$V_i = \frac{1}{Det(a)} \left\{ \sum_{i=1}^{k} C_{ji}\ S_i\ p \sum_{j=1}^{k} C_{ji}\ n_j \right\} for\ i \le 1\ \le k, (4)$$

Where, $\{c_{ij} \mid j=1, 2, \ldots, k \ \&\ i=1, 2, \ldots, k\}$ is the $(j, i)^{th}$ cofactor for matrix {a}; $\{n_j \mid j=1, 2, \ldots, k\}$ is the set of integer values used to compensate the mod operation used in equation (3).

Since share works on segmenting image file into several files, it is an appropriate that the image compression starts, the need for data compression as a topic acquired its importance because it is a solution key to pypass the insufficient storage space and limited bandwidth for data transmission[8][9]. The used programs to compress images are, in fact, employ designed techniques that exploit unimportant sensory information and statistical redundancies. Most of images programmers rely on the use of two techniques (i.e., sub-band coding and transform coding). Sub-band coding decomposes signal into a number of sub-bands, using band-pass filter like wavelet transform [10]. Transform coding uses a mathematical transformation like DCT and FFT. The mathematical representation for Two-dimensional DCT is:

1. The forward 2D DCT equation is:

$$G_{ij} = \sqrt{\frac{4}{nm}}\ C_i C_j \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} p_{xy}\ \cos\left[\frac{(2y+1)j\pi}{2m}\right] \cos\left[\frac{(2x+1)i\pi}{2n}\right], (5)$$

For $0 \le i \le n-1$ and $0 \le j \le m-1$ and for Ci and Cj

1. The inverse 2D DCT equation is:

$$p_{xy} = \sqrt{\frac{4}{nm}} C_i C_j \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} C_i C_j G_{ij} \cos\left[\frac{(2x+1)i\pi}{2n}\right] \cos\left[\frac{(2y+1)j\pi}{2m}\right], (6)$$

For $0 \le x \le n-1$ and $0 \le y \le m-1$

Where,

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$$

An emergence interest was grown in the recent years about utilizing the benefits of wavelet coding to process both images and audio applications. The concept of wavelet coding, like other transform coding techniques is based on the idea that the coefficients of transform decorrelates the samples values of the signal such that they can be coded in more compressive way comparing with the case of direct compression of the original samples values themselves. The following set of equations describes the four "lifting" steps and the two "scaling" steps that are applied to accomplish the bi-orthogonal (9/7) wavelet decomposition:

$$Y_{2n+1} = X_{2n+1} + a\ X_{2n} + X_{2n+2}\ , \quad (7)$$
$$Y_{2n} = X_{2n} + b\ Y_{2n-1} + Y_{2n+1}\ , \quad (8)$$
$$Y_{2n+1} = Y_{2n+1} + c\ Y_{2n} + Y_{2n+2}\ , \quad (9)$$
$$Y_{2n} = Y_{2n+1} + d\ Y_{2n-1} + Y_{2n+1}\ , \quad (10)$$
$$Y_{2n+1} = -K\ Y_{2n+1}\ , \quad (11)$$
$$Y_{2n} = (1/K)\ Y_{2n}\ , \quad (12)$$

Where, X() is the input array & Y() is the wavelet transform outcome array (i.e., the approximation & detail coefficients). The values of the (a, b, c, d, K) parameters are:

a=-1.58613434, b=-0.0529801185, c=-0.8829110762 d=-0.4435068522, K=1.149604398 [11].

Developing a safer image secret sharing scheme is still considered difficult task except that there is a need to take away the following drawbacks found within the earlier accessible secret sharing schemes [12]:

- **a.** They need high computational complexity throughout each sharing and revealing stages.
- **b.** Reducing the shares size (i.e., huge shadow size).
- **c.** High consumption of network bandwidth.
- **d.** A compromise of encryption key.
- **e.** Secret information hacking.
- **f.** Shares modification.
- **g.** For compression, developing a system can get the benefits of existing spectral redundancy in the input image. The system should use the proper mapping equations to generate uncorrelated color image bands convey low data content.
- **h.** Decomposing image signal into sub bands, each one conveys a certain part of the signal that has specific spectral characteristic.
- **i.** Pruning the existing local spatial correlation by using transform coding.

Developing a proper set of entropy encoders to efficiently prune the existing statistical redundancy may found in the transformed data.

2- RELATED WORK

Gornale et al. (2007) suggested a compression system based on the bi-orthogonal wavelet filters; because orthogonal filters have suitable property of energy preservation whereas bi-orthogonal filters lack of it. Since, Daubechies, Symlet and Coiflet filters have better property of energy conservation, more vanishing moments, regularity and asymmetry than other orthogonal filters; they were adopted in their suggested system. Also, they used bi-orthogonal wavelet filter out of Daubechies, Symlet and Coiflet for lossy fingerprint image compression. They have applied Daubechies, Symlet and Coiflet Wavelet Transforms (WT) through different orders at 1 to 5 decomposition levels on the fingerprint images [13].

Singh et al. (2011) referred that the properties of wavelet transform greatly help in identifying and selecting of significant and non-significant coefficients amongst the wavelet coefficients, DWT represents image as a sum of wavelet functions (wavelets) in different resolution levels. So, the basis of wavelet transform can be composed of functions satisfy the requirements of multi-resolution analysis. So, the choice of wavelet function for image compression depends on the image application and the content of the image. They presented a review of the fundamentals of image compression based on wavelet. Also, they discussed important features of wavelet transform in compression of images. Finally, they evaluated and compared the compression performance of three different wavelet families (i.e., Daubechies, Coiflets, Biorthogonal) through measuring the image fidelity objectively (using peak signal-to-noise ratio) and subjectively (using visual image quality), beside to compression ratio [14].

Ruchika et al. (2012) they proposed the use of DWT to compress wide variety of medical images. They indicated

that the application of thresholds on DWT coefficients in addition to Huffman encoding leads to major reduction in image statistical redundancy [6][14]. Test results indicated that the system performance is promising when it is applied on medical images [15].

Shaymaa et al. (2015) have proposed a lossy compression scheme used different signal representation method. Firstly, they used cubic Bezier surface (CBI) representation to prune image component that shows large scale variation. The produced cubic Bezier surface is subtracted from the image signal to get the residue component. Then, bi-orthogonal wavelet transform was applied to decompose the residue component [5][13]. Finally they used some lossless coding method to boost the compression gain [16].

Mustafa Ulutas et al. (2011) proposed a medical image is distributed among a number of clinicians in telediagnosis and each one of them has all the information about the patient's medical condition. However, disclosing all the information about an important patient's medical condition to each of the clinicians is a security issue. They proposed a (k, n) secret sharing scheme which shares medical images among a health team of n clinicians such that at least k of them must gather to reveal the medical image to diagnose. Shamir's secret sharing scheme is used to address all of these security issues in one method. The proposed method can store longer EPR strings along with better authenticity and confidentiality properties while satisfying all the requirements as shown in the results [17].

Laith et al. (2012) refereed that a Pixel expansion and bad image quality are the most cons of Visual Secret Sharing approach. They have proposed two visual secret sharing techniques for grayscale images to overcome these drawbacks, one is based on Fast Fourier Transform (FFT) and the other is based on incorporating Discrete Wavelet Transform (DWT) with FFT. In the first technique, FFT is used for image sharing while in the second technique; DWT and FFT are employed for image sharing effectively. In each visual secret sharing technique, every share is encrypted using Multilayer Security Method (MSM). Also, some image compression concepts were used to eliminate the pixel expansion problem which is produced before the secret sharing process [18].

3- PROPOSED IMAGE COMPRESSION SYSTEM

Like any typical image compression schemes, the introduced system consists of two individual units: (i) image compressor is used to compress real color still images with little rate of subjective distortion, and (ii) image decompressor (reconstructor) is used to retrieve the raster image. In the following sections details for the implied stages of each unit are given.

3.1 Lossy Image Compressor

As shown in figure (1) this unit consists of the following stages:

3.1.1 Image Loading: Initially, the input image is loaded as bitmap (raster) formatted. Then, the loaded image data is analyzed to the basic color components (RGB).

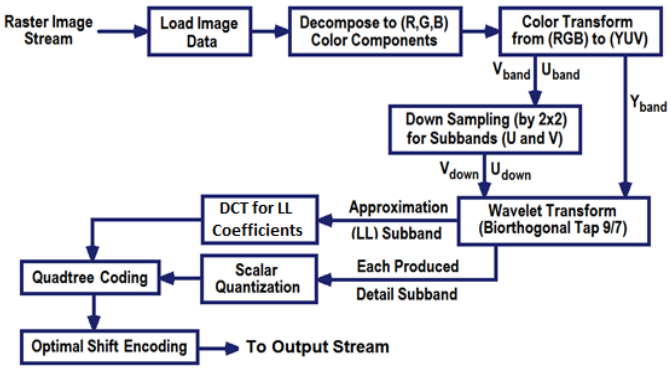

Figure 2: The layout of proposed lossy image compressor

3.1.2 Color Transform: The advantage of dealing with the color representation YUV is to get proper image data representation that is closer to performance nature of Y Human Vision System (HSV); the intensity band (i.e., Y) is the most subjective, informative channel of the color image; while the bands U and V normally convey less subjective information.

3.1.3 Chromatic Bands Downsampling: Since the chrominance components (U and V) holds only 10% of the whole image information and HSV doesn't have high spatial resolution against these band; so these bands are down sampled by 2 to produce the down sampled components (i.e., U_{down} & V_{down}). This down sampling will cause insignificant subjective distortions in the color image.

3.1.4 Biorthogonal Wavelet Transform: In this paper, the bi-orthogonal wavelet transform (tap 9/7) is used as a spatial signal decomposer for each subband, individually. Bi-orthogonal wavelet decomposition was chosen due to its compressive efficiency, and modern wide use in standards lossless & lossy compression scheme (for example in ISO JPEG2000 standard). In wavelet transform coding the image is divided into four subbands each one can easily be encoded separately.

The bio-orthogonal tap 9/7 wavelet filters are applied to the (Y, U_{down} and V_{down}) color bands separately. The transform will decompose the data of each colors band into to four subbands (i.e., LL, LH, HL and HH).

3.1.5 DCT: The approximation subband (LL) is passed through the DCT transform coding. The process of Passing the $LL_{Subband}$ throw the DCT is illustrated in figure (2) shown below.

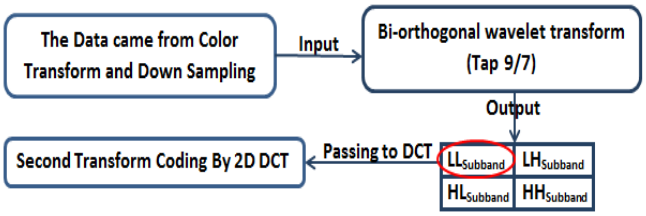

Figure 1: The process of passing the $LL_{Subband}$ values

3.1.6 Quantization: Quantization is simply the process of reducing the number of bits that are needed to store the values of coefficients by reducing their accuracy, the main objective of quantization is to reduce the high-frequency coefficients (least importance) to zero. In the proposed system, the uniform scalar quantization operation was adopted to quantize the coefficients of each sub band

individually; this step will reduce the number of needed bits to represent the coefficients approximately, and preparing it to the shift coding step. The coefficients of each subband are quantized with an appropriate quantization step value (Qstp). The transform coefficients are categorized according to its subband membership to (L_n, H_n… H_2, H_1). The rounded subband (L_n) coefficients are quantized using low quantization step, which is always smaller than the quantization step used to quantize the detail subbands' coefficients. Also, the quantization step of the high level detail subband coefficients is smaller than that for low level subband.

In the proposed system, a hierarchal relationship was adopted to determine the value of scalar quantization step that is used to separately quantize the wavelet (i.e., detail) coefficients belong to each sub band. The way of selecting the variation nature of quantization step across the subbands was based on the criteria "diminishing the range of wavelet coefficients values without making significant degradation in image quality". The adopted hierarchal scalar quantization step was governed by the following equation:

$$Q_{step}(n) = \begin{cases} Q\alpha^{n-1} & \text{for LH and HL subbands} \\ Q\beta\alpha^{n-1} & \text{for HH subbands} \end{cases} \quad (13)$$

Where, $Q_{step}(n)$ is the quantization step of n^{th} subband; α is the rate of increasing in quantization step and its value should be less than 1; β is the additional ratio for quantization step for HH sub band and its value is always (≥1).

3.1.7 Quadtree Coding: In this step, the process of quadtree coding is applied to encode the quantized detail bands (i.e., LH, HL & HH) subbands of (Y, U_{down}, and V_{down}). The quadtree method divides the subband into four equal size square blocks. Then, each block is tested to check if it has at least non-zero coefficient value (i.e., not empty) or not (i.e., empty). In case the tested block is not empty it will be divided into four sub-blocks, and made the search process on the sub-blocks (4x4) is retested alone. This process begins with whole subband quantized coefficients and stops when reaching to the smallest sub-blocks of size (2x2); if that block is not empty then its 4 coefficients values are stored in a temporary buffer (Buf()) along with the quadtree partitioning binary code.

3.1.8 Entropy Encoding Using Shift Coding: The last step of the proposed image compressor unit is applying lossless compression of the non-empty blocks coefficients which are already registered in temporary buffer Buf(). Figure (3) illustrates the layout of the developed entropy coder; which is designed to remove the statistical redundancy efficiently.

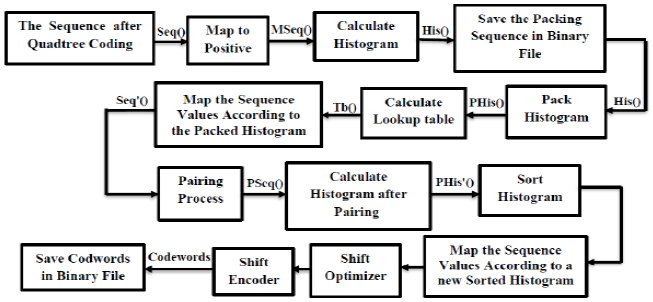

Figure 3: The Layout of the Enhanced Shift-key Encoder

The histogram of data sequence registered in Buf() is concentrated around the high peak located at the value (0). Therefore, the use of shift coding will be appropriate for the entropy encoding the data to attain high compression gain. The implanted enhanced shift coder implies the following steps:

(a) Mapping to Positive: This step maps the sequence elements values to be positive numbers. This step makes the coding process of the next step easier. This mapping step is done by representing each negative element value as a positive odd number, and each positive value is represented as an even number. This conversion can be applied using the following simple mapping equation:

$$m_{out}(i) = \begin{cases} 2m_{in}(i) & if\ m_{in}(i) \geq 0 \\ -2m_{in}(i) + 1 & if\ m_{in}(i) < 0 \end{cases} \quad (14)$$

Where, $m_{in}(i)$ is the mapped i^{th} sequence element; $m_{out}(i)$ is the corresponding mapped value.

(b) Histogram Packing: The histogram of the positive sequence elements will show long tail with many long gaps places (i.e., many values don't occur in the sequence); this leads to significant reduction in the compression gain when using shift encoder. So, it is proper to make a compacting in the histogram values. Then, the positive buffer's elements values are remapped according to the packed histogram elements.

(c) Elements Pairing Stage: This step is applied to detect the most redundant pair of subsequent elements, and replace the pair by a single value (i.e., Max+1), where Max is the highest registered value of all elements. After each pairing step, Max value is incremented by 1. The pairing operation can be repeated for a number of times (M); where M is a predefined parameter value.

After the Pairing step, the new histogram of produced sequence is calculated and stored as overhead information in compression stream, because it is necessary in decoding operation. Then, it is sorted in descending order and the elements values are re-mapped according to the indexes of their values in the histogram.

(d) Determination of Optimal Shift Key Value: According to shift key encoding mechanism, the small values symbols which have high occurrence probabilities are assigned short codewords, while the large values symbols are assigned long codewords [10]. So, for determining the optimal key value that separating the short codewords from the long ones, an optimizer algorithm was introduced. This optimizer searches for the best short codeword length (n_S) and the corresponding long codeword (n_L) leads to the lowest value from the total bits (T_{Bits}) required to encode the whole input symbols; that is:

$$T_{bits} = n_S \sum_{i=0}^{2^S-1} His(i) + n_L \sum_{i=2^S-1}^{M} His(i) \quad (15)$$

Where, His(i) is the i^{th} histogram value; M is the highest value of the input elements to the shift encoder.

(e) Shift Encoding: As last step, the traditional shift encoding step is applied such that the small coded element is encoded by using the leading bit value "0" concatenated with n_S bits used to represent the element value; while the large value element is encoded by

using the leading bit value "1" concatenated with n_L bits used to represent the element value.

4- IMAGE SHARING SYSTEM

4.1 Double Stream Ciphering

This stage is applied to prune by encrypting the correlation may appear and/or the bits significance problem that may exist in the created sharing coefficients. This module consists of two sub-modules, the first one is to do the preliminary stage that aims to prepare the random sequences {$Seq_i(0..L_i)$ | i=1 to 4}and the initial values for their associated counters {Lc_i | i=1 to 4}. The second sub-module is a function generates a random number at each call epoch.

The outcomes of the random generator are firstly used to encipher the generated compression data. For this stage the random sequence is initiated using a group ciphering key that is known for all shareholders. Secondly, each produced share is encrypted individually; such that the random generator is reinitiated using the privates shareholder encryption. The XOR convolution process is applied to encipher the plaintext.

4.1.2 Initialization of Random Sequence

In this sub-module four steps are applied to initialize four rotating random sequences. The length of each one of these sequences (i.e., L_1, L_2, L_3, L_4) is a prime number, in such case the overall periodicity of the produced random numbers will be the sum of product of the sequences length (i.e., $L_1xL_2xL_3xL_4$). The four steps on this sub-module are:

Step1: Checking the length of pre-assigned ciphering key, its length should be not less than 6. Also, for ensuring the assigned key doesn't convey high level of symmetry. it is concatenated with a string has a good randomness attributes.

Step2: Calculating the initial values of the 16-bits numbers (K_1, K_2, K_3, and K_4) using the contents of the pre-assigned ciphering key; the calculation depends on the following formula:

$$K_i = \sum_{j}^{N} (K_i + Sec[j] * P_i)\ Mod\ 65536 \quad (16)$$

Where, Sec() is the value of ciphering key; n is the length of it; P_i is a prime number that can be selected arbitrarily; i is an index, its values are i=1,2,3,4.

Step3: Generating the binary sequence {$A_1()$, $A_2()$, $A_3()$, $A_4()$} depending on the calculated K's values by the following relationship:

$$if\ K_i\ and\ 2^j = 0\ then\ A_i[j] = 1\ else\ A_i[j] = 0 \quad (17)$$

Where, i={1,2,3,4} is the index of the sequence, j={0,1,…,7} is the index of the calculated bit.

Step4: In this step, the four random bytes sequences {$Seq_i()$ | i=1,2,3,4} are generated using modified linear Congruential model, as follows:

$$Seq_i(k) = \sum_{j=0}^{15} (M_i(k) + A_i[j](Q_{i0} + R_i\, j)\ mod\ 65536 \quad (18)$$

$$M_i(k) = M_{i,0} + Stp_i \times i \quad (19)$$

Where, Q_{i0}, R_i, $M_{i,0}$ & Stp_i are arbitrarily selected prim numbers.

Step5: As the last step the initials values of the sequence counters (Lc_1, Lc_2, Lc_3, and Lc_4) are determined using the linear Congruential model:

$$M = \sum_{j=0}^{15} \left(M + A_i(j) \times (B_i + D_i \times j)\right) \wedge 65536 \quad (20)$$

$$Lc_i = M\ mod\ L_i \quad (21)$$

The values of (B_1, B_2, B_3, B_4), and (D_1, D_2, D_3, D_4) are taken arbitrarily as prime numbers. The initial value of M is taken 0.

The above steps indicate the confusion condition is satisfied because any bit change applied on the cipher key will lead to a complete change in the random sequences.

4.1.3 Generation of Random Number

This process is achieved by convoluting the numbers picked from the four rotating sequences; such that at each call of random number generation calls a number is picked from each sequence and XORed with the numbers belong to other sequences. So, a new random number is determined by convoluting the four numbers extracted from the sequences {$Seq_i()$}:

$$R = Seq_1[Lc_1] \wedge Seq_2[Lc_2] \wedge Seq_3[Lc_3] \wedge Seq_4[Lc_4] \quad (22)$$

Where, {Lc_1, Lc_2, Lc_3, Lc_4} are the counters values, their values should be bounded such that (Lc_i=L_i | 0<i<5); the values {L_1, L_2, L_3, L_4} are the lengths of random sequences, that are at each call, the counters are updated using:

$Lc_i = (Lc_i+1)\ mod\ L_i$ *for i=1, 2, 3, 4* (23)

4.2 Shares Generation Stage

The stage of shares generation was designed to confirm the generated shares are as secure as possible and have low size. This module consists of two sub-modules: (i) the proposed (k, n) threshold scheme, and (ii) identical proposed encryption algorithm which is employed for creating XOR convolution, but used an individual key for encrypting every share severally. Figure (3.10) shows the fundamental steps of the applied shares generation stage. The details of these sub modules are illustrated in the next sections.

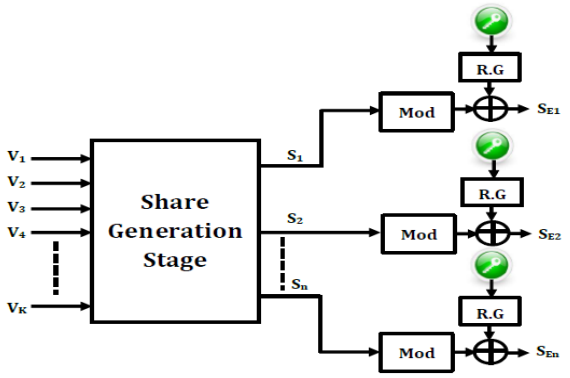

Figure 4: Share generator stage

4.2.1 The Proposed (k, n) Threshold Scheme

The proposed ***(k, n)*** threshold algorithm relies on Blakley's secret sharing methodology. It uses the principle of hyperplane geometry to resolve the secret sharing problem. According to this scheme, the secret is a point in a k dimensional space; that is the point of intersection of all the hyperplanes equations (1, 2 and 3).

Due to using a prime number larger than the range of secret codewords (i.e., byte or parts of it); the output of equation (3.15) covers the range [0, p-1] that exceeds the secret codeword range. For illustration, just in case the codewords are bytes, their range is [0,255], the closest prime

number (***p***) that's higher than byte range is 257; in such case, one of the possible outputs of equation (3.14) is 256 which needs additional bits (9 bits) rather than 8 bits (which are required to represent the input values). In order to handle this overshot value the following two Alternative ways will be chosen:

- **(i)** Either by increasing the number of bits used to represent the input codewords by 1 so as to represent the output codewords (i.e., assigning 9 bits for all outcomes of equation (3);
- **(ii)** Or using key-shift method to bypass this small extension within the dynamic range of the input and output codewords.

The following equations present the percentage of increase within the size output/ input code words because of using the two codewords' bits illustration methods:

$$\Delta^{+}_{bitsext} = \frac{1}{L} \times 100\% \quad (24)$$

$$\Delta^{+}_{keyshift} = \frac{P - (2^{l} - 1)}{P} \times 100\% \quad (25)$$

Where, ***L*** is the number of bits used to represent the input secret code words; ***p*** is the used primary number. Table (3.2) shows the percentage value for various input codewords size; it is obvious that the key-shift technique causes smaller increase in comparison with the bits expansion methodology. For this reason the key-shift methodology is adopted to handle the overshot values of the share output.

Table 1: the percentage of increase in bits/code word used to represent one share value (output) relative to secret value (input)

Code Word Size L (in bits)	*P*	$\Delta_{BitsExt}$	$\Delta_{KeyShift}$
8	257	12.500	0.778
7	131	14.286	3.053
6	67	16.667	5.970
5	37	20.000	16.216
4	17	25.000	11.765

4.2.2 Load the Binary Sequence

As first step, the output of the compression process is loaded as a binary sequence. This loaded data will be considered the input file to the sharing process. First, the length data is allocated; then the file content is loaded as one bulk of data and stored in one-dimensional array.

4.2.3 Encryption the Secret Sequence Using Group Key

At first the user enters a password (*Group Secret Key* which is the first level of data security). Here, there is a possibility that the password (Group Key) be acceptable but weak. In order to avoid this situation a random key is added, this extra key is added using a hash function. But in the case that a password (*Group Key*) is very weak (i.e., its length is less than 6 characters) system will reject the key and ask for using a proper *Group Key*.

After shares generation stage same encryption algorithm is used to encrypt each share separately. The difference is only in the used secret key.

5- IMAGE REVEAL UNIT

The reveal unit consists of the inverse operations to those applied within the sharing process; and these operations are applied in reverse order. The reveal unit consists of the four basic modules: (i) shares assembler, (ii) diffusion recovery, (iii) decompression, and (iii) post processing (filtering)). All these modules consist of many sub modules; each one performs certain specific tasks.

5.1 Shares Assembler

For reveal purposes, the users ought to consider that they have to get a particular threshold number (i.e., k) of shares. These shares ought to be decrypted, and so fed to share assembler module. The module consists of 2 sub modules: (i) shares decryption and (ii) a (k, n) threshold reveal scheme.

5.1.1 Shares Decryption

Shares decryption is the first step in shares assembler module, the secret information of every used share needs to be decrypted. This needs generating the same random sequence that utilized in the second encryption (i.e., individual encryption) stage; this random sequence will be used to accomplish each XOR convolution followed by random permutation.

5.1.2 The (k, n) Threshold Revealing Scheme

The revealing part of (k, n) threshold scheme is the inverse of the sharing part that used equation (3).

Due to the missing values of {n()}, the retrieval of each set {V()| i=1, 2,..,k} requires testing a large number of n's trials, which in turn makes the reveal process by applying equation (3) is too costly, especially when k is large (i.e., k>2).

As another reveal solution, another method is proposed, the method takes the advantages of advents occurred in the size of the used volatile memories in personal computers. The proposed mechanism uses large memory allocation to determine k number of arrays, known as Tbl(); each one has a size p^k, which covers all possible combinations of share values. The elements of the first array, $Tbl_1()$, holds the corresponding first secret value, the second array holds the corresponding second secret value… etc. Table (3.3) presents the required memory size to do revealing using the second method (i.e., memory based method) just in case of taking p=257 (i.e., the smallest prime number exceeds the byte range). The table shows that for cases k > 3 the required memory allocation reaches multi-Tera byte size which isn't possible. To handle this memory allocation problem the range of representing the secret information is reduced to be smaller than 8 bits (i.e., byte representation), this can facilitate in reducing the value of p that is appropriate for preserving the required memory; table (3.4) shows the required memory size for the cases of k > 2 and for different secret codewords representations.

Algorithm (3.15) presents the steps of applying the quick (second) method.

Table 2: the list of required memory allocation for different K values and p=257

K	*Required Memory*
2	132098 Bytes = 129 Kbytes
3	50923779 Bytes = 48.565 Mbytes
4	17449881604 Bytes = 16.251 Tbytes
5	5.60577E+12 Bytes = 5220.784 TBytes
6	1.72882E+15 = 1610089.881 Tbytes

Table 3: the list of required memory allocation for different k and p values

K	Codeword Size	P	The Required Memory Size			
			Bytes	Kbytes	Mbytes	Tbytes
3	7	131	6.74E6	6586	6.432	0.006
	6	67	902289	881	0.860	0.001
	5	37	151959	148	0.145	0.000
	4	17	14739	14.4	0.014	0.000
4	7	131	1.17E9	1.15E6	1123	1.097
	6	67	8.06E7	78715	76.870	0.075
	5	37	7.49E6	7321	7.149	0.007
	4	17	334084	326	0.319	0.000
5	7	131	1.93E11	1.88E8	183961	180
	6	67	6.75E9	6.59E6	6438	6.287
	5	37	3.47R8	338594	332	0.323
	4	17	7.1E6	6933	6.770	0.007
6	7	131	3.03E13	2.96E10	2.89R7	28241
	6	67	5.43E11	5.3E8	5.18E5	505
	5	37	1.54E10	1.50E7	14681	14.34
	4	17	1.44E8	1.41E5	138	0.135

6- IMAGE DIFFUSION RECOVERING STAGE

The diffusion recovery stage has similar steps as those mentioned in Diffusion stage (see section 4) the difference is only within the order that's applying XOR permutation step before the random permutation step.

7- IMAGE DECOMPRESSOR UNIT

Figure (5) presents the layout of the Image Decompressor. The order of the corresponding inverse operations (to those used in image compressor) is arranged in the reverse order.

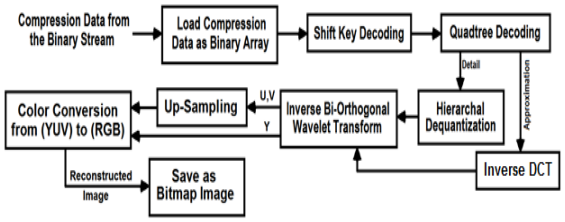

Figure 5: Decoding Process

Also, figure (6) presents the layout of the corresponding shift key decoder.

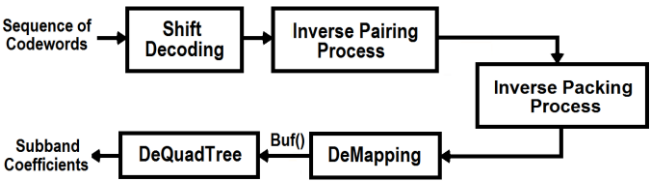

Figure 6: Entropy Decoder

8- Results and Discussion

Different sets of tests have been performed to evaluate the performance of the proposed color image compression system in terms of Compression ratio (CR) and Peak Signal to Noise ratio (PSNR). The effectiveness of the following system parameters was investigated:

(1) The number of wavelet passes (Npass).

(2) Initial quantization step (Q) for the detail subbands coefficient (LH, HL, and HH).

(3) The descending rate parameter (α).

(4) Beta increment ratio (β).

Table (4) lists the specifications of the standard images used as a test material in this paper work. Table (5) presents the default values for the investigated parameters.

Table 4: the Characteristics of the Tested Standard Images

Characteristic	Color Lena	Color Barbara	Gray Barbara
Bit depth (bit)	24	24	8
Dimension	256x256	256x256	256x256
Size (KB)	192	192	192

Table 5: the Default Values of the Control Parameters for all images

Parameter	Default Value		Range
	Color Lena, Barbara	Gray Barbara	
QS_Y	35	35	[10,50]
$QS_{UD,VD}$	35	35	
α_y	0.5	0.5	[0.1,0.9]
$\alpha_{UD,VD}$	0.5	0.5	
β_y	1.6	1.6	[1.1,1.9]
$\beta_{UD,VD}$	1.6	1.6	
N_{pass}	3	3	[1,4]
BS_{Size}	16	16	[2,32]

Table (6), (7) and (8) list the attained performance parameters of the proposed system when applied on color Lena, Barbara and gray Barbara, respectively using different number of passes; the listed results indicate an acceptable compression results can be reached in terms of (CR), fidelity measures and (PSNR) as in each test group in the tables.

In all conducted tests, the values of three parameters were fixed and the value of the fourth parameter was changed to define its effect on the compression system performance. The results listed in above tables indicate the following remarks:

1. In general, an increase in CR is occurred when the values of the parameters (number of Passes**,** Q_{step}**,** **α,** **β**) are increased, all an associated decrease occurred in PSNR and an increase in MSE.
2. The parameter Q_{step} is the most effective parameter on the compression performance; its increase causes significant increase on CR and decrease in PSNR**.**
3. The parameter No. of Passes has significant impact on compression performance; its increase shows an increase in CR and decrease in PSNR. But in general it is less effective in comparison with Q_{step}.
4. The parameter **β** is less effective on CR, PSNR and MSE.
5. Generally, the value of MSE is increased when the value of any one of the four parameters is increased. But, this increase varies according to the parameter type; **α** is the largest dependency and **β** is the least dependency.

Table 6: Test Results for Color Lena image

NoPass	BS	Alpha	Beta	Qstep	CR	PSNR	MSE	Time (in seconds)	
								Encode	Decode
1	16	0.5	1.6	35	9.80	31.05	51.01	0.57	0.58
2	16	0.5	1.6	35	18.23	28.82	85.15	0.29	0.17
3	16	0.5	1.6	35	24.24	27.72	109.79	0.25	0.08
4	16	0.5	1.6	35	23.75	27.26	122.19	0.27	0.06
3	2	0.5	1.6	35	26.36	27.74	109.22	0.24	0.05
3	4	0.5	1.6	35	24.94	27.74	109.22	0.24	0.05
3	8	0.5	1.6	35	25.08	27.74	109.39	0.28	0.05
3	16	0.5	1.6	35	25.40	27.72	109.79	0.25	0.10
3	32	0.5	1.6	35	25.50	27.68	110.86	0.34	0.17
3	16	0.5	1.6	35	29.08	27.72	109.79	0.24	0.08
3	16	0.6	1.6	35	31.37	26.84	134.59	0.25	0.07
3	16	0.7	1.6	35	36.38	26.20	155.86	0.24	0.07
3	16	0.8	1.6	35	42.61	25.60	179.06	0.25	0.08
3	16	0.9	1.6	35	48.59	25.08	201.64	0.22	0.08
3	16	0.5	1.5	35	28.46	27.75	109.11	0.24	0.09
3	16	0.5	1.6	35	25.78	27.72	109.79	0.25	0.07
3	16	0.5	1.7	35	25.93	27.67	110.94	0.36	0.07
3	16	0.5	1.8	35	26.07	27.64	111.87	0.28	0.07
3	16	0.5	1.9	35	26.40	27.61	112.58	0.28	0.08
3	16	0.5	1.6	30	21.07	28.23	97.71	0.30	0.07
3	16	0.5	1.6	35	24.80	27.72	109.79	0.33	0.07
3	16	0.5	1.6	40	29.00	27.20	123.87	0.25	0.07
3	16	0.5	1.6	45	33.23	26.74	137.52	0.24	0.07
3	16	0.5	1.6	50	36.93	26.44	147.44	0.24	0.08

Table 7: Test Results for Color Barbara Image

NoPass	BS	Alpha	Beta	Qstep	CR	PSNR	MSE	Time (in seconds)	
								Encode	Decode
1	16	0.5	1.6	35	8.78	28.43	93.26	0.56	0.61
2	16	0.5	1.6	35	14.67	26.79	136.07	0.40	0.19
3	16	0.5	1.6	35	18.44	25.88	167.79	0.32	0.07
4	16	0.5	1.6	35	18.11	25.53	181.88	0.32	0.05
3	2	0.5	1.6	35	19.95	25.89	167.42	0.28	0.05
3	4	0.5	1.6	35	18.85	25.89	167.42	0.28	0.05
3	8	0.5	1.6	35	19.00	25.88	167.58	0.37	0.07
3	16	0.5	1.6	35	19.12	25.88	167.79	0.32	0.07
3	32	0.5	1.6	35	19.15	25.87	168.12	0.38	0.18
3	16	0.5	1.6	35	21.88	25.88	167.79	0.27	0.08
3	16	0.6	1.6	35	22.24	25.28	192.48	0.28	0.07
3	16	0.7	1.6	35	24.90	24.67	221.35	0.37	0.08
3	16	0.8	1.6	35	27.58	24.12	251.67	0.32	0.07
3	16	0.9	1.6	35	29.76	23.68	278.335	0.24	0.08
3	16	0.5	1.5	35	21.39	25.92	166.22	0.30	0.08
3	16	0.5	1.6	35	19.40	25.88	167.79	0.30	0.08
3	16	0.5	1.7	35	19.61	25.83	169.741	0.33	0.08
3	16	0.5	1.8	35	20.06	25.77	171.84	0.28	0.07
3	16	0.5	1.9	35	20.51	25.73	173.49	0.32	0.08
3	16	0.5	1.6	30	15.05	26.47	146.41	0.33	0.07
3	16	0.5	1.6	35	18.51	25.88	167.79	0.29	0.07
3	16	0.5	1.6	40	22.16	25.38	188.03	0.26	0.07
3	16	0.5	1.6	45	26.12	24.99	206.04	0.25	0.07
3	16	0.5	1.6	50	30.33	24.63	223.89	0.23	0.07

Table 8: Test Results for gray Barbara Image

NoPass	BS	Alpha	Beta	Qstep	CR	PSNR	MSE	Time (in seconds)	
								Encode	Decode
1	2	0.5	1.6	35	40.81	37.92	10.48	0.40	0.25
2	2	0.5	1.6	35	22.62	36.47	14.65	0.41	0.06
3	2	0.5	1.6	35	15.92	36.08	16.02	0.37	0.04
4	2	0.5	1.6	35	13.02	35.99	16.33	0.36	0.03
1	2	0.5	1.6	35	16.33	36.09	15.96	0.30	0.03
1	4	0.5	1.6	35	15.05	36.09	15.98	0.30	0.03
1	8	0.5	1.6	35	15.25	36.08	16.02	0.42	0.03
1	16	0.5	1.6	35	15.35	36.04	16.14	0.37	0.08
1	32	0.5	1.6	35	15.38	35.97	16.44	0.38	0.26
1	2	0.5	1.6	35	18.06	35.30	19.17	0.36	0.03
1	2	0.6	1.6	35	21.27	34.47	23.18	0.31	0.03
1	2	0.7	1.6	35	24.37	33.72	27.57	0.322	0.03
1	2	0.8	1.6	35	27.43	32.88	33.49	0.30	0.03
1	2	0.9	1.6	35	30.39	32.06	40.44	0.30	0.03
1	2	0.5	1.5	35	15.58	36.01	16.27	0.34	0.035
1	2	0.5	1.6	35	15.93	35.93	16.59	0.32	0.038
1	2	0.5	1.7	35	16.29	35.87	16.79	0.33	0.033
1	2	0.5	1.8	35	16.66	35.80	17.09	0.36	0.039
1	2	0.5	1.9	35	17.03	35.74	17.31	0.32	0.034
1	2	0.5	1.6	10	6.43	40.75	5.46	0.67	0.04
1	2	0.5	1.6	20	10.62	37.12	12.593	0.46	0.03
1	2	0.5	1.6	30	17.43	35.13	19.94	0.29	0.03
1	2	0.5	1.6	40	26.25	32.73	27.50	0.26	0.03
1	2	0.5	1.6	50	36.09	32.80	34.12	0.22	0.03

Table 9: lists the compression results when the DCT method is applied on Lena image; we can notice the effectiveness of applying DCT on the proposed system which gets high CR and better quality (PSNR). The results show clearly an increase in CR is occurred when DCT is applied; also insignificant change in PSNR and MSE is also occurred.

Table 9: Test Results for Color Lena image with and without DCT implementation

Without implementation DCT					
No. Pass	CR	PSNR	MSE	Time (in seconds)	
				Encode	Encode
1	8.11	29.78	68.28	0.47	0.04
2	15.41	28.35	94.92	0.25	0.03
3	25.77	27.38	118.62	0.31	0.04
4	33.40	26.86	133.86	0.24	0.04
With implementation DCT					
1	26.53	29.92	66.20	0.55	0.58
2	41.11	28.45	92.81	0.32	0.17
3	45.77	27.46	116.53	0.25	0.08
4	50.06	19.97	653.63	0.23	0.05

Figure (7) shows samples of reconstructed image produced by the proposed DCT method with the original image for color Lena and part of it.


Original


Reconstructed


Original


Reconstructed

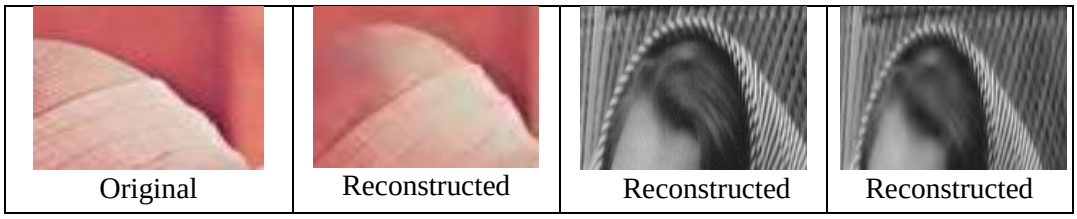

Figure 7: Samples of Compression Result

Table 9: The best compression results when DCT was applied

Image	*Qs*	*α*	*β*	*BSIZ*	*NPass*	*CR*	*PSN*
Color Lena	35	0.2	1.9	16	2	41.25	30.08
	35	0.4	1.9	16	3	58.75	28.39
Color Barbara	25	0.4	1.9	16	2	34.13	28.15
	35	0.3	1.9	16	3	37.17	27.01
Gray	35	0.9	1.9	16	3	105.8	30.25

Table 10: The Compression File Size and the Size of Share File in term of CR and PSNR for the images list in table

Image	No pass	CR	PSNR	Compression file Size	Share File Size
Color Lena	3	83.7	27.29	4 KB	2 KB
Color Lena	2	58.0	29.62	5 KB	2.5
Color Lena	2	43.2	30.23	7 KB	3.5
Color Barbara	3	39.1	27.10	8 KB	4 KB
Color Barbara	3	30.3	28.073	10 KB	5 KB
Color Barbara	1	19.1	30.13	14 KB	7 KB
Color Pepper	2	45.2	27.55	6 KB	3 KB
Color Pepper	3	40.6	27.13	8 KB	4 KB
Color Pepper	3	21.2	29.40	13 KB	6.5
Gray Barbara	3	63.0	31.55	2 KB	1 KB
Gray Barbara	3	34.3	33.11	3 KB	1.5

Table (10) lists Sharing results where appear the size of one share for all images, this will lead to increased compression ratio by half, which reduces file size and speed up the transfer process

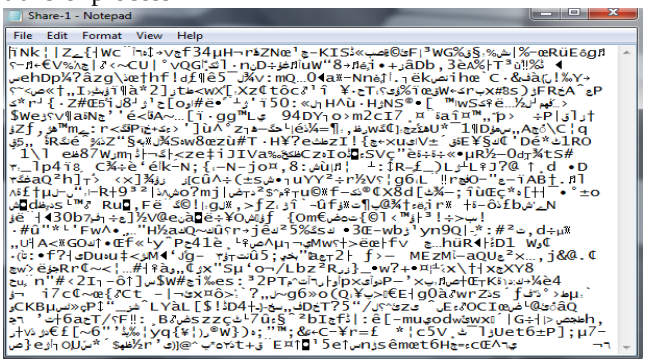

Figure 8: Samples of the content of two share files for Lena color image

Figure 9: Samples of the content of two share files for Lena color image

9- COMPARISONS WITH PREVIOUS STUDIES

Many methods for image compression have been developed in the past few years. In this section the results of our proposed schemes have been compared with some previously published methods.

Table (11) lists the CR and PSNR attained by our proposed schemes with those given in previous studies, with taking into consideration that in these studies the same images have been used. The listed results demonstrated that our proposed scheme outperforms other methods.

Table 11: Comparison between the performance results of several image compression methods, (Not Mentioned (NM))

Author	*Image*	*CR*	*PSNR (in dB)*	*Bit Depth (in Bit)*	*Size (in KB)*
[19]	Lena	4.72	NM	8	64
	Barbara	3.7	NM	8	64
[20]	Lena	5.76	31.5	8	64
[21]	Lena	12.05	26.89	24	64
[22]	Lena	13.6	30.12	24	192
[23]	Lena	8.0	NM.	8	64
	Barbara	7.0	NM.	8	64
[24]	Lena	5.412	NM.	8	64
[25]	Lena	19.08	30.04	24	192
	Lena	6.09	31.35	8	64
	Barbara	11.80	30.07	24	192
	Barbara	5.06	30.18	8	64
[25]	Lena	16.16	30.12	24	192
	Lena	6.02	32.15	8	64
	Barbara	10.75	30.01	24	192
	Barbara	5.01	31.49	8	64

Proposed scheme DCT	Lena	**19.15**	**34.84**	**24**	**192**
	Barbara	**11.75**	**33.92**	**24**	**192**
	Lena	**6.10**	**42.77**	**8**	**64**
	Barbara	**5.04**	**41.84**	**8**	**64**

The results show that the proposed system is competitor with the standard compression schemes and methods introduced in the literature, So that when we use the same compression ratio (CR), we could get an image quality (PSNR) higher than obtained in the previous studies. The results show that the proposed system was able to compete the previous studies strongly.

d. As a future work, using image fractal coding as compression tool (instead of DCT and/or Wavelet transform coding) in the compressed image.

10- CONCLUSION

In this paper, an image compression scheme based on using DCT, wavelet, Quadtree and high order shift coding had been introduced. The following remarks are stimulated:

a. The use of DCT had improved the compression performance (i.e., increase the CR while preserving the image quality).
b. The increase in quantization step causes an increase in compression ratio and a decrease in PSNR value.
c. Q_{step} is the most effective parameter on compression performance; while the parameter β is the less effective one.

REFERENCES

[1] Naskar, P. K.; Khan, H. N.; Roy, U.; Chaudhuri, A.; and Chaudhuri, A.; ***"Shared Cryptography with Embedded Session Key for Secret Audio"***, International Journal of Computer Applications, ISSN: 0975 – 8887, Vol. 26, No. 8, Pp. 5-9 July 2011.

[2] Beimel, A.; ***"Secure Schemes for Secret Sharing and Key Distribution"***, Ph.D. Dissertation, Senate of Technion, Israel Institute of Technology, June 1996.

[3] Beim, A.; ***"Secret-Sharing Schemes: A Survey"***, Coding and Cryptology, Lecture Notes in Computer Science Vol. 6639, Pp. 11-46, 2011.

[4] Iftene, S.; "Secret Sharing Schemes with Applications in Security Protocols", Ph.D. Dissertation, University "Alexandru Ioan Cuza" of Iasi, Romania, 2007.

[5] Nikam, A.; Kapade, P.; and Patil, S.; ***"Audio Cryptography: A (2, 2) Secret Sharing for Wave File"***, International Journal of Computer Science and Application, ISSN: 0974-0767, Issue 2010, Pp. 96-99, 2010.

[6] Martin, R.; ***"Introduction to Secret Sharing Schemes",*** 2008.

[7] Bozkurt, I. N.; Kaya, K.; Selcuk, A. A.; and Guloglu, A. M.; ***"Threshold Cryptography Based on Blakley Secret Sharing"***, In Proc. of Information Security and Cryptology 2008, Ankara, Turkey, Dec 2008.

[8] P. Havaldar, G. Medioni; **2004** "Multimedia Systems Algorithms Standards and Industry Practices", Book, Cengage Learning, Boston, MA, USA, 2010.Salomon, D.; "Data Compression: The Complete Reference"; Book; Springer; New York.

[9] D. Salomon; **2004** "Data Compression: The Complete Reference", Book, Springer; New York, Fourth Edition.

[10] D. Katz, R. Gentile; **September-2005**, "Embedded Media Processing", Book; Elsevier Science, ISBN 978-0-7506-7912-1, 432 Pages.

[11] D.C. Dhubkarya and S. Dubey; **2009**, "High Quality Audio Coding at Low Bit Rate Using Wavelet and Wavelet Packet Transform"; Journal of Theoretical and Applied Information Technology, Vol. 6, No. 2, Pp. 194-200.

[12] Yeshwantrao, S.A.; Jadhav, V.J.; and Rahate, P.S.; ***"Shared Cryptographic Scheme with Efficient Data Recovery and Compression for Audio Secret Sharing"***, International Journal of Emerging Technology and Advanced Engineering, ISSN: 2250-2459, Vol. 4, Issue 2, Pp. 408-414, February; 2014.

[13] S.S. Gornale, R.R. Manza, V. Humbe and K.V. Kale, **January 2007** "Performance Analysis of Bi-Orthogonal Wavelet Filters for Lossy Fingerprint Image Compression", International Journal of Imaging Science and Engineering (IJISE), ISSN: 1934 9955, Vol. 1, No .1, Pp. 16-20.

[14] Priyanka Singh, Priti Singh, R. K. Sharma; **January 2011**, "JPEG Image Compression based on Bi-Orthogonal, Coiflets and Daubechies Wavelet Families", International Journal of Computer Applications, ISSN: 0975 – 8887, Vol. 13, No.1, Pp. 1-7.

[15] Ruchika, M. Singh, Anant. R. S; **May 2012** "Compression of Medical Images Using Wavelet Transforms" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Vol. 2, Issue 2, Pp. 339-343,

[16] S.D. Ahmed, L.E. George and Ban N. Dhannoon, **2015** "The Use of Cubic Bezier Interpolation, Biorthogonal Wavelet and Quadtree Coding to Compress Color Images", British Journal of Applied Science & Technology, DOI: 10.9734/BJAST/ 2015/20480, Vol. 11, Issue 4, Pp. 1-11.

[17] Mustafa Ulutas, Güzin Ulutas, Vasif V. Nabiyev; ***"Medical image security and EPR hiding using Shamir's secret sharing scheme"***, Dept. of Computer Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey, The Journal of Systems and Software 84 (2011) 341–353,ELSEVIER.

[18] Laith Hamid Abed **" Development of Visual Secret Sharing Techniques "** M.Sc. Thesis, Collage of Science, University of Anbar (2012).

[19] El-Harby, A. A.; and Behery, G. M.; "*Qualitative Image Compression Algorithm Relying on Quadtree*", CGST-GVIP,Vol. 8, No. 3, October 2008.

[20] Siddeq, M. M.; "*Using Shift Number Coding with Wavelet Transform for Image Compression*", Journal of Information and Computing Science**,** Vol. 4, No. 3, Pp. 311-320, 2009.

[21] Krishnaiah, G. C.; Jayachandraprasad, T.; and GiriPrasad, M.N.; "Efficient Image Compression

Algorithms Using Evolved Wavelets", *International Journal Systems and Technologies*, Vol. 4, No. 2, pp. 127-146, 2011.

[22] George, L. E.; and Sultan, B. A.; "*The Use of Bi-Orthogonal Wavelet, 2D Polynomial and Quadtree to Compress Color Images*"; Signal Processing, Image Processing and Pattern Recognition (SIP-2011), Communications in Computer and Information Science (CCIS), Springer-Verlag Berlin Heidelberg, Vol. 260, pp. 379- 390, 2011.

[23] Gupta, K.; and Verma R. L.; "*Minimum entropy based Lossless Image Compression using Predictive Coding and Integer Wavelet Transform*", International Journal of Engineering Science and Innovative Technology (IJESIT), Vol. 2, No. 4, pp. 603-612, July 2013.

[24] Al-Khafaji, G.; "*Image Compression based on Quadtree and Polynomial* ", International Journal of Computer Applications, Vol. 76, No. 3, pp. 31 - 37, August 2013.

S.D. Ahmed; "*Image Compression using adaptive polynomial transform*", M.Sc. Thesis , College of Science, Baghdad University, Iraq, 2016.